

## HOMEWORK FOR BACKEND DEVELOPER POSITION IN BSC

The submitted code should be of the quality we can expect during a normal working week. Using maven or ant is not a requirement; however candidates are welcome to do so. Candidates should provide instructions on how to run the application from the command line. As candidates will not have the opportunity to clarify requirements, they are advised to note any assumptions in their submission. However candidates are welcome to use any external libraries necessary to complete the tasks it is not important to include big number of libraries and platforms "just to demonstrate I can use them". Code should be as clear and effective as possible.

There are no tricks or gotchas: only implement the functionality described below. Code quality is as important as solving the problem itself.

### Exercises

#### Payment Tracker

Write a program that keeps a record of payments. Each payment includes a currency and an amount. Data should be kept in memory (please don't introduce any database engines).

The program should output a list of all the currency and amounts to the console once per minute. The input can be typed into the command line with possibility to be automated in the future, and optionally also be loaded from a file when starting up.

Sample input:

```
USD 1000
HKD 100
USD -100
RMB 2000
HKD 200
```

Sample output:

```
USD 900
RMB 2000
HKD 300
```

Detailed requirements:

When your Java program is run, a filename can be optionally specified. The format of the file will be one or more lines with Currency Code Amount like in the Sample Input above, where the currency may be any uppercase 3 letter code, such as USD, HKD, RMB, NZD, GBP etc. The user can then enter more lines into the console by typing a currency and amount and pressing enter. Once per minute, the output showing the net amounts of each currency should be displayed. If the net amount is 0, that currency should not be displayed.

When the user types "quit", the program should exit.

You may need to make some assumptions. For example, if the user enters invalid input, you can choose to display an error message or quit the program. For each assumption you make, write it down in a readme.txt and include it when you submit the project.

Things you may consider using:

- Unit testing



- Threadsafe code
- Documentation
- Programming patterns

Please put your code in a bitbucket/github repository. We should be able to build and run your program easily (you may wish to use Maven, Ant, etc). Include instructions on how to run your program.

Optional bonus question

Allow each currency to have the exchange rate compared to USD configured. When you display the output, write the USD equivalent amount next to it, for example:

USD 900

RMB 2000 (USD 314.60)

HKD 300 (USD 38.62)

